

Socket Programming

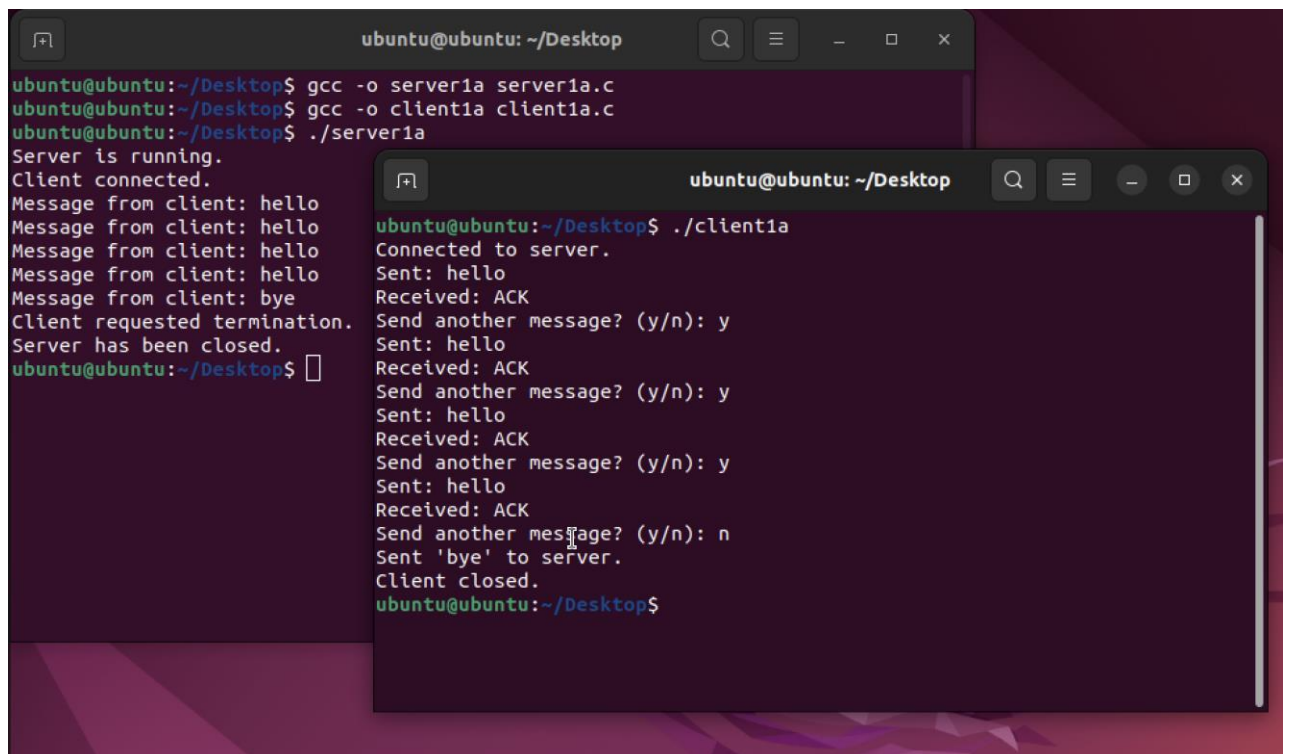
Q1: Concurrent Server Problem

(A) Develop a client and a server program using socket interface as follows: client sends a “hello” message and wait till “ACK” is received. Server prints the “hello” message and sends an ACK.

(B) Modify the program of (A) such that the client will block after sending five “hello” messages. The window slides by one unit after receiving each ACK.

Outputs for Q1-A and Q1-B:

1A:



The screenshot shows two terminal windows. The left window is the server's output, and the right window is the client's output.

Server Output (Left Window):

```
ubuntu@ubuntu: ~/Desktop
ubuntu@ubuntu:~/Desktop$ gcc -o server1a server1a.c
ubuntu@ubuntu:~/Desktop$ gcc -o client1a client1a.c
ubuntu@ubuntu:~/Desktop$ ./server1a
Server is running.
Client connected.
Message from client: hello
Message from client: hello
Message from client: hello
Message from client: hello
Message from client: bye
Client requested termination.
Server has been closed.
ubuntu@ubuntu:~/Desktop$
```

Client Output (Right Window):

```
ubuntu@ubuntu: ~/Desktop
ubuntu@ubuntu:~/Desktop$ ./client1a
Connected to server.
Sent: hello
Received: ACK
Send another message? (y/n): y
Sent: hello
Received: ACK
Send another message? (y/n): y
Sent: hello
Received: ACK
Send another message? (y/n): y
Sent: hello
Received: ACK
Send another message? (y/n): n
Sent 'bye' to server.
Client closed.
ubuntu@ubuntu:~/Desktop$
```

1B:

```
ubuntu@ubuntu: ~/Desktop  ubuntu@ubuntu: ~/Desktop
ubuntu@ubuntu:~/Desktop$ gcc -o server1b server1b.c
ubuntu@ubuntu:~/Desktop$ gcc -o client1b client1b.c
ubuntu@ubuntu:~/Desktop$ ./server1b
Server listening.
Client connected.
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: bye
Client initiated termination.
Client connected.
Received message: hello
Received message: hello

ubuntu@ubuntu:~/Desktop$ ./client1b
Connected to server.
Sent message #1: hello
Sent message #2: hello
Sent message #3: hello
Sent message #4: hello
Sent message #5: hello
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Sent message #6: hello
Sent message #7: hello
Sent message #8: hello
Sent message #9: hello
Sent message #10: hello
Received ACK: ACK
Received ACK: ACK
Sent message #11: hello
Sent message #12: hello
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Sent message #13: hello
Sent message #14: hello
Sent message #15: hello
Received ACK: ACK
Received ACK: ACK
Sent message #16: hello
Sent message #17: hello
Received ACK: ACK
Received ACK: ACK

Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: bye
Client initiated termination.
Client connected.
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: bye
Client initiated termination.

Received ACK: ACK
Sent message #16: hello
Sent message #17: hello
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Sent message #18: hello
Sent message #19: hello
Sent message #20: hello
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Sent 'bye' to server.
Client closed.
ubuntu@ubuntu:~/Desktop$ ./client1b
Connected to server.
Sent message #1: hello
Sent message #2: hello
Sent message #3: hello
Sent message #4: hello
Sent message #5: hello
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Sent message #6: hello
Sent message #7: hello
Sent message #8: hello
Received ACK: ACK
Received ACK: ACK
Sent message #9: hello
Sent message #10: hello
```

```
ubuntu@ubuntu: ~/Desktop
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: bye
Client initiated termination.
Client connected.
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: hello
Received message: bye
Client initiated termination.
[ ]

ubuntu@ubuntu: ~/Desktop
Sent message #8: hello
Received ACK: ACK
Received ACK: ACK
Sent message #9: hello
Sent message #10: hello
Received ACK: ACK
Sent message #11: hello
Received ACK: ACK
Sent message #12: hello
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Sent message #13: hello
Sent message #14: hello
Sent message #15: hello
Sent message #16: hello
Received ACK: ACK
Received ACK: ACK
Sent message #17: hello
Sent message #18: hello
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Sent message #19: hello
Sent message #20: hello
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Received ACK: ACK
Sent 'bye' to server.
Client closed.
ubuntu@ubuntu: ~/Desktop$
```

Q2: Chat Program

Modify the chat program in such a way that the server and the client can involve in two-way chatting (i.e., Full duplex communication).

Output for Q2:

```
ubuntu@ubuntu: ~/Desktop
ubuntu@ubuntu:~/Desktop$ gcc -o server2 server2.c
ubuntu@ubuntu:~/Desktop$ gcc -o client2 client2.c
ubuntu@ubuntu:~/Desktop$ ./server2
Server is running.
Client connected.
Client: hello server! is the connection established?
hello server!
Server: hello server!
yes it is established!
Server: yes it is established!
Client: great!
Client disconnected.
ubuntu@ubuntu:~/Desktop$ ./server2
Server is running.
Client connected.
hello
Server: hello
Client: hi
bye
Server: bye
Server has disconnected.
ubuntu@ubuntu:~/Desktop$

ubuntu@ubuntu:~/Desktop$ ./client2
Connected to server.
hello server! is the connection established?
Client: hello server! is the connection established?
Server: hello server!
Server: yes it is established!
great!
Client: great!
bye
Client: bye
Client has disconnected.
ubuntu@ubuntu:~/Desktop$ ./client2
Connection to server failed: Connection refused
ubuntu@ubuntu:~/Desktop$ ./client2
Connected to server.
Server: hello
hi
Client: hi
Server disconnected.
ubuntu@ubuntu:~/Desktop$
```